FIGURE 1

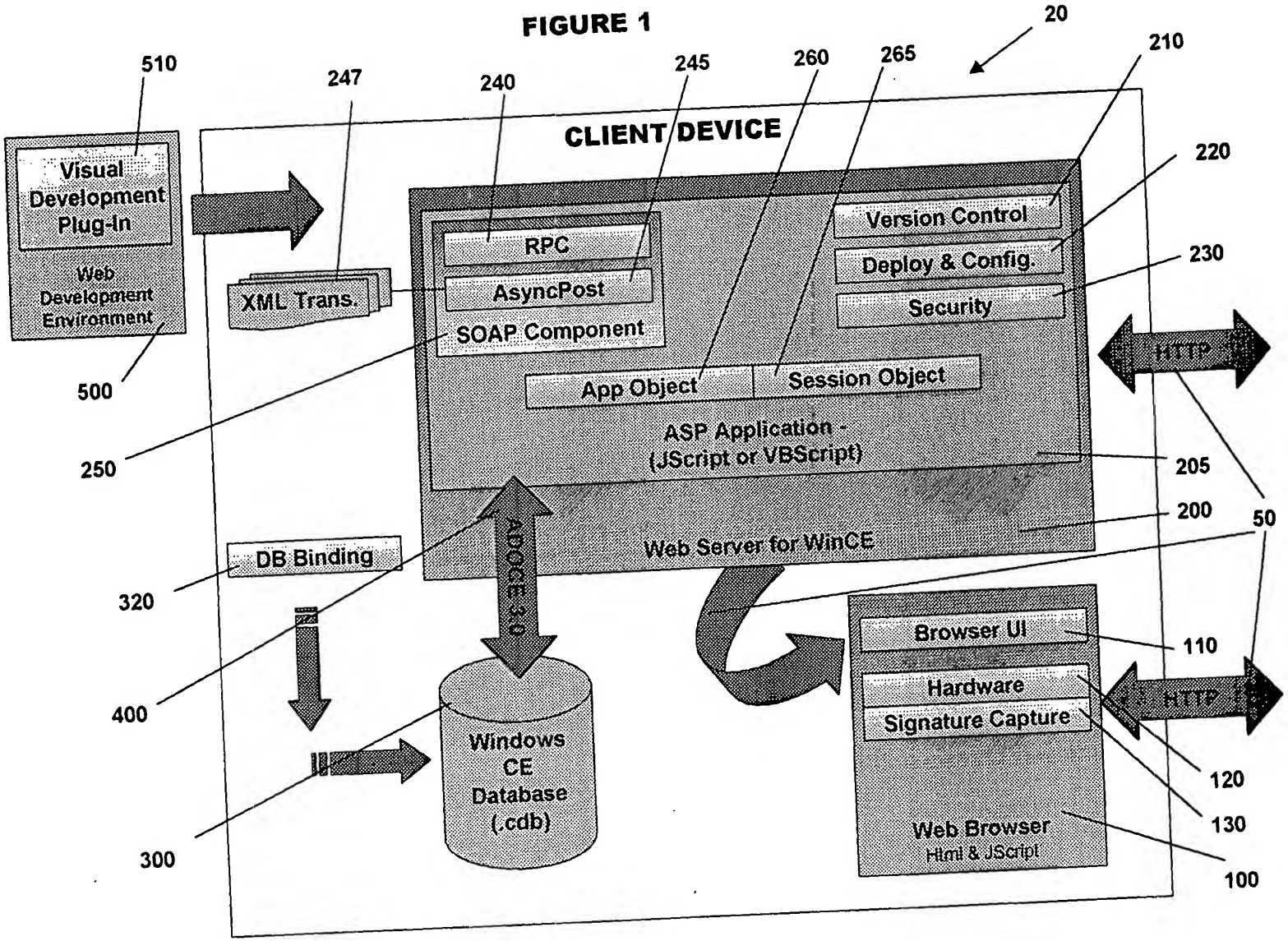

20
510
247
240
245
260
265
210
CLIENT DEVICE
Visual Development Plug-In
Web Development Environment
500
XML Trans.
RPC
AsyncPost
SOAP Component
250
Version Control
Deploy & Config.
Security
220
230
HTTP
App Object
Session Object
ASP Application - (JScript or VBScript)
205
200
50
Web Server for WinCE
DB Binding
320
ADOCE 3.0
400
Windows CE Database (.cdb)
300
Browser UI
110
Hardware
Signature Capture
HTTP
120
130
Web Browser
Html & JScript
100

FIGURE 2

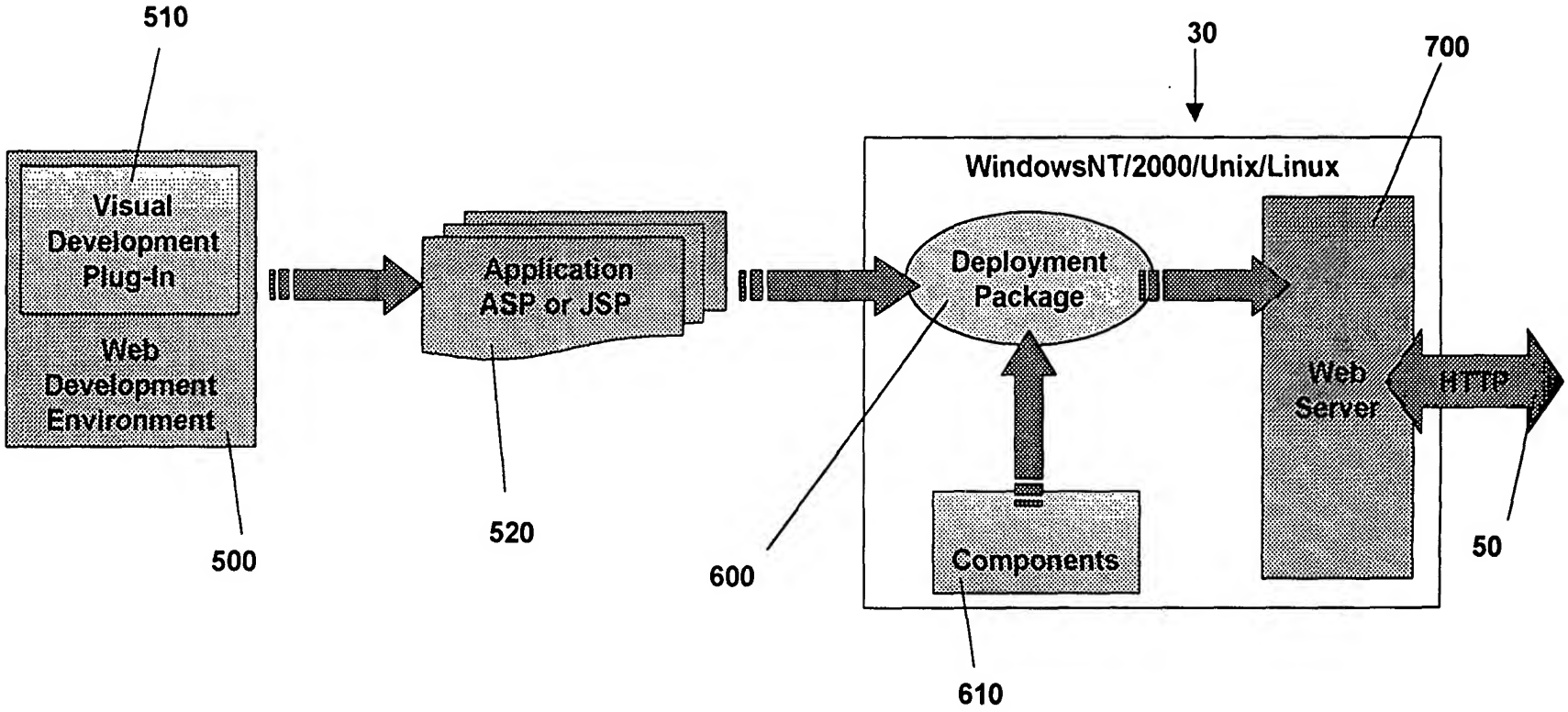

510
Visual Development Plug-In
Web Development Environment
500
Application ASP or JSP
520
30
WindowsNT/2000/Unix/Linux
Deployment Package
600
Components
610
700
Web Server
HTTP
50

FIGURE 3

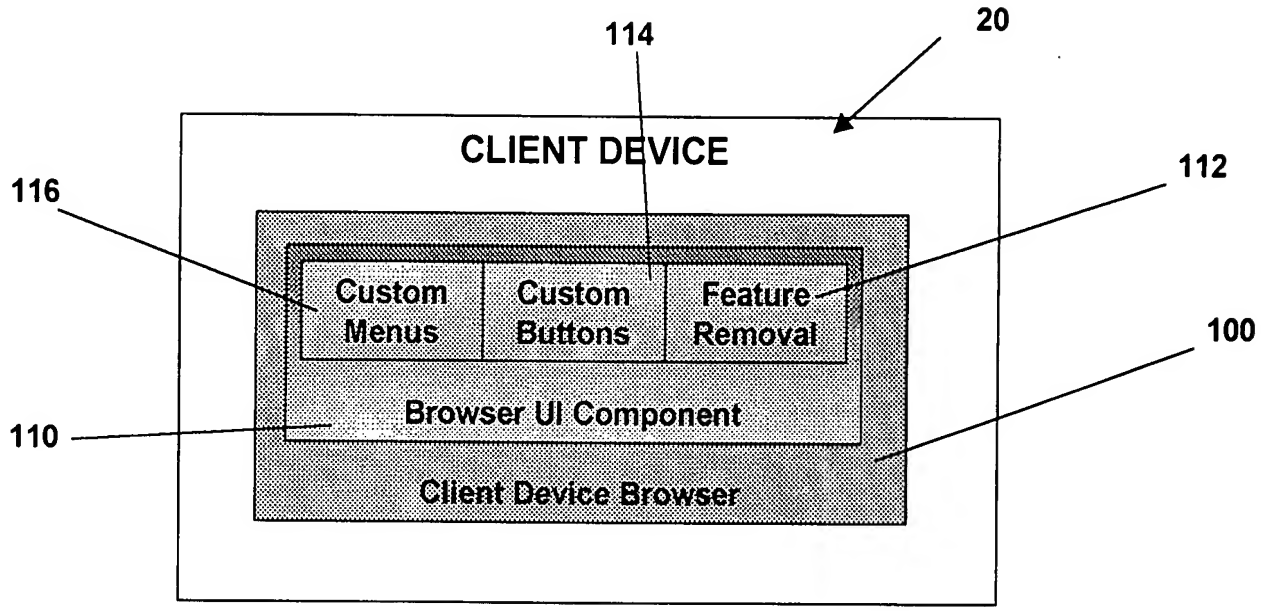

114
20
CLIENT DEVICE
116
112
Custom Menus
Custom Buttons
Feature Removal
100
110
Browser UI Component
Client Device Browser

FIGURE 4

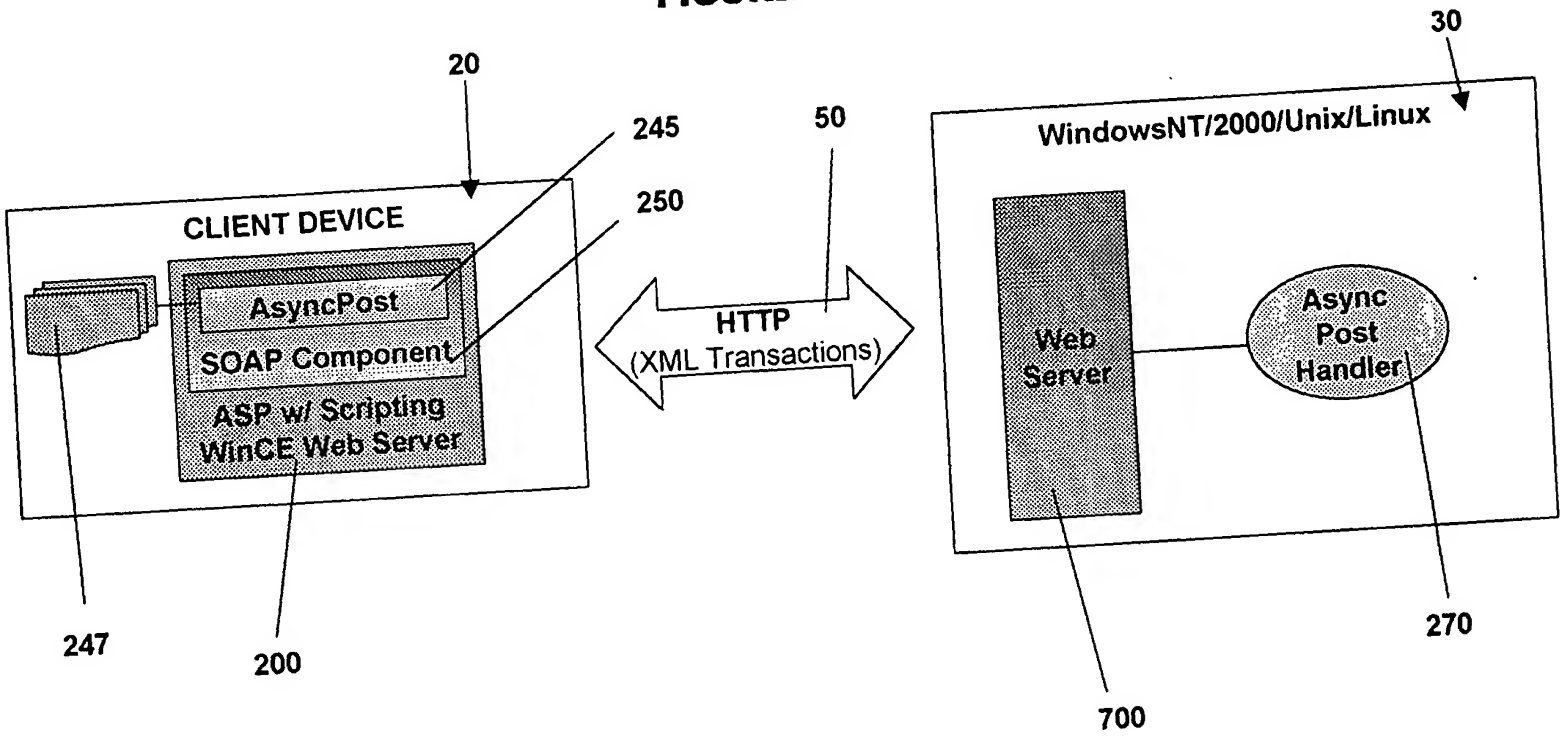

20
245
50
30
WindowsNT/2000/Unix/Linux
CLIENT DEVICE
250
AsyncPost
SOAP Component
ASP w/ Scripting
WinCE Web Server
HTTP
(XML Transactions)
Web
Server
Async
Post
Handler
247
200
700
270

FIGURE 5

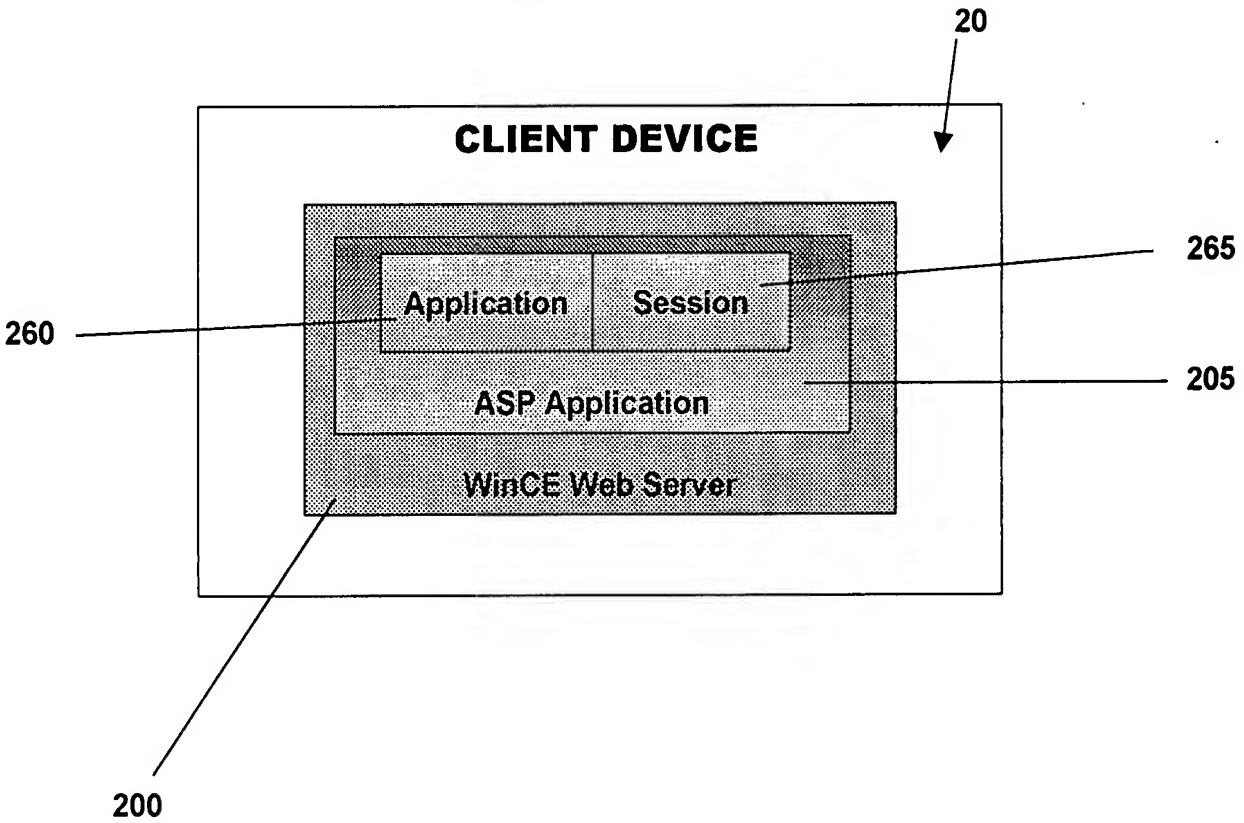

20
CLIENT DEVICE
265
Application
Session
260
205
ASP Application
WinCE Web Server
200

FIGURE 6

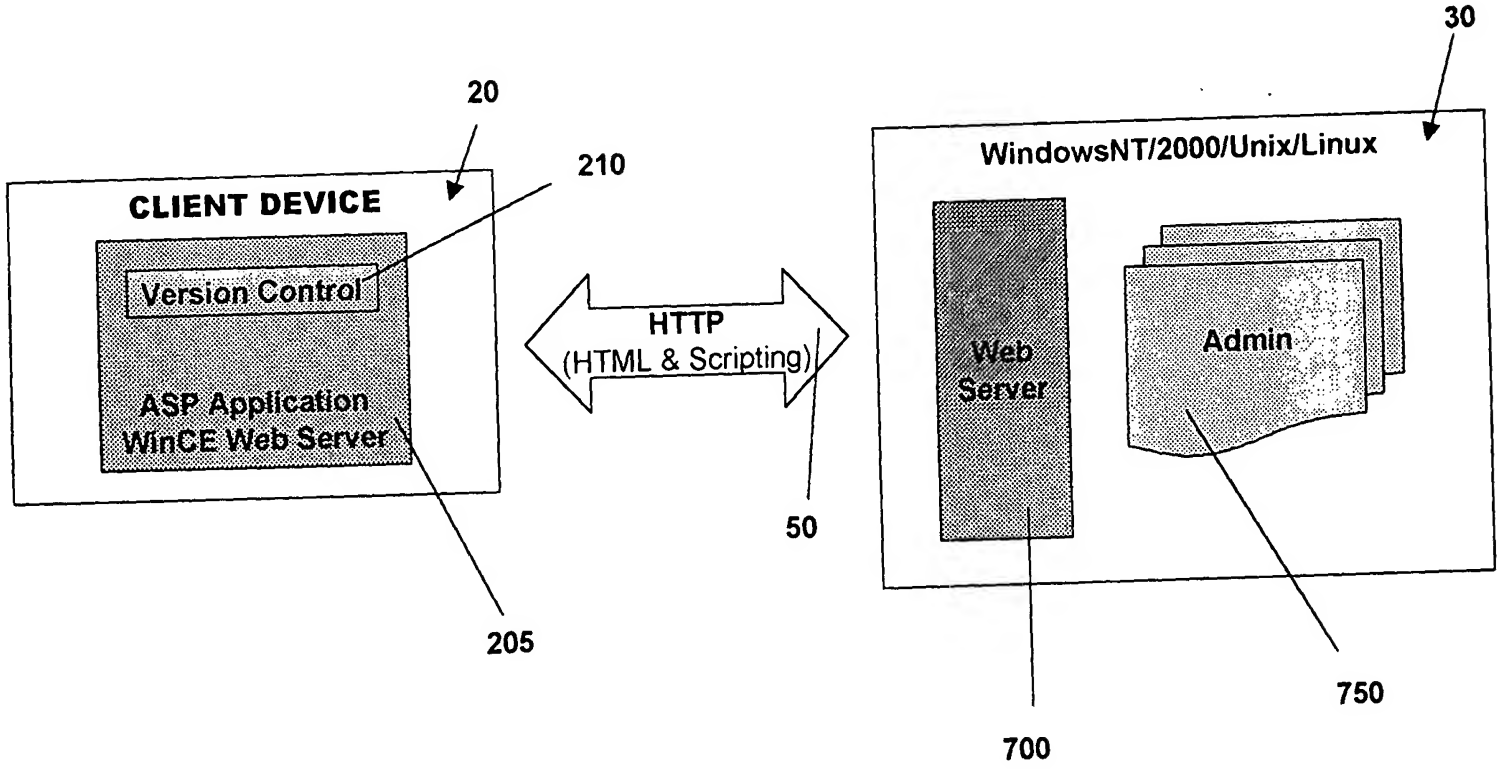

30
20
210
CLIENT DEVICE
Version Control
ASP Application
WinCE Web Server
HTTP
(HTML & Scripting)
WindowsNT/2000/Unix/Linux
Web
Server
Admin
50
205
750
700

FIGURE 7

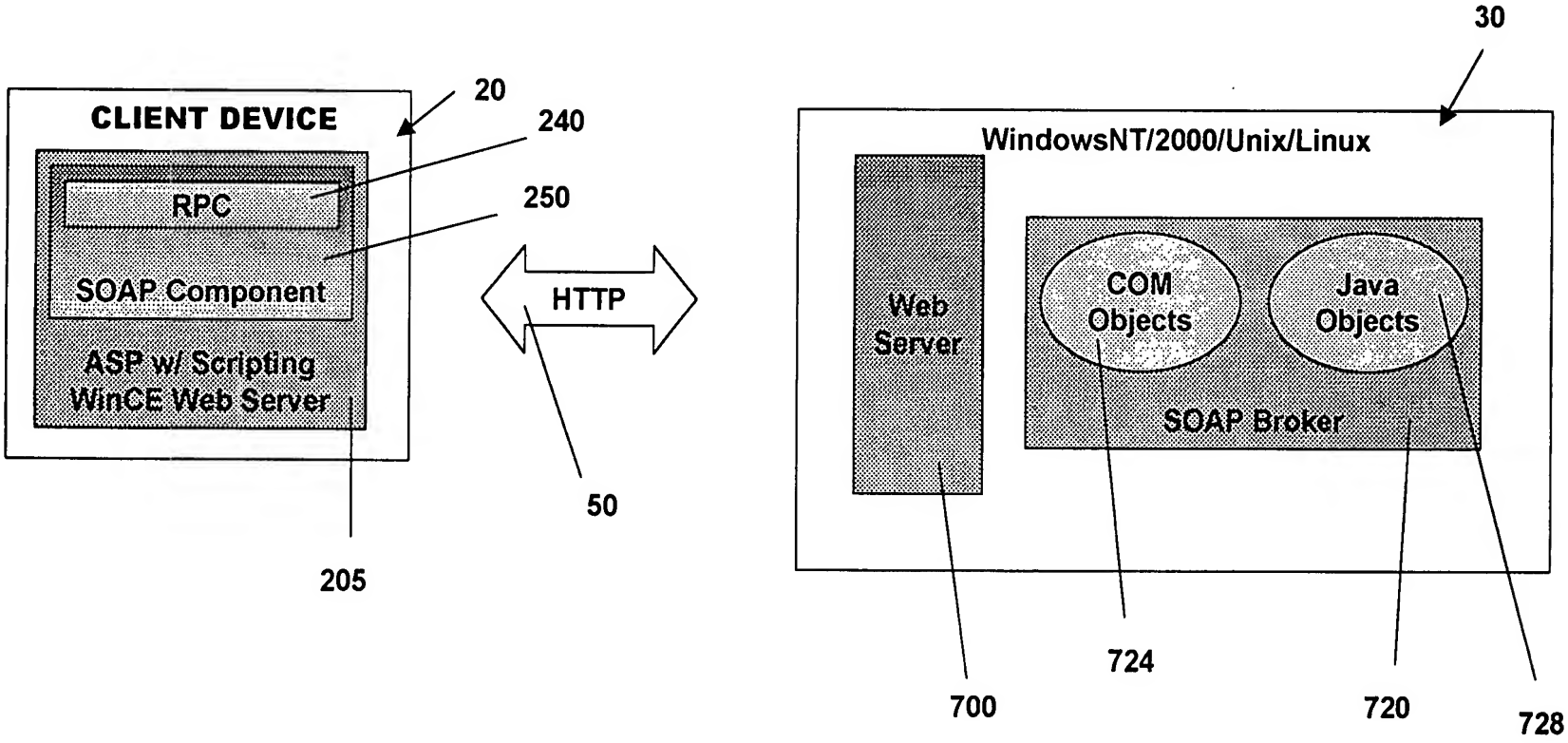

CLIENT DEVICE
20
240
RPC
250
SOAP Component
ASP w/ Scripting
WinCE Web Server
HTTP
50
205
30
WindowsNT/2000/Unix/Linux
Web
Server
COM
Objects
Java
Objects
SOAP Broker
700
724
720
728

FIGURE 8

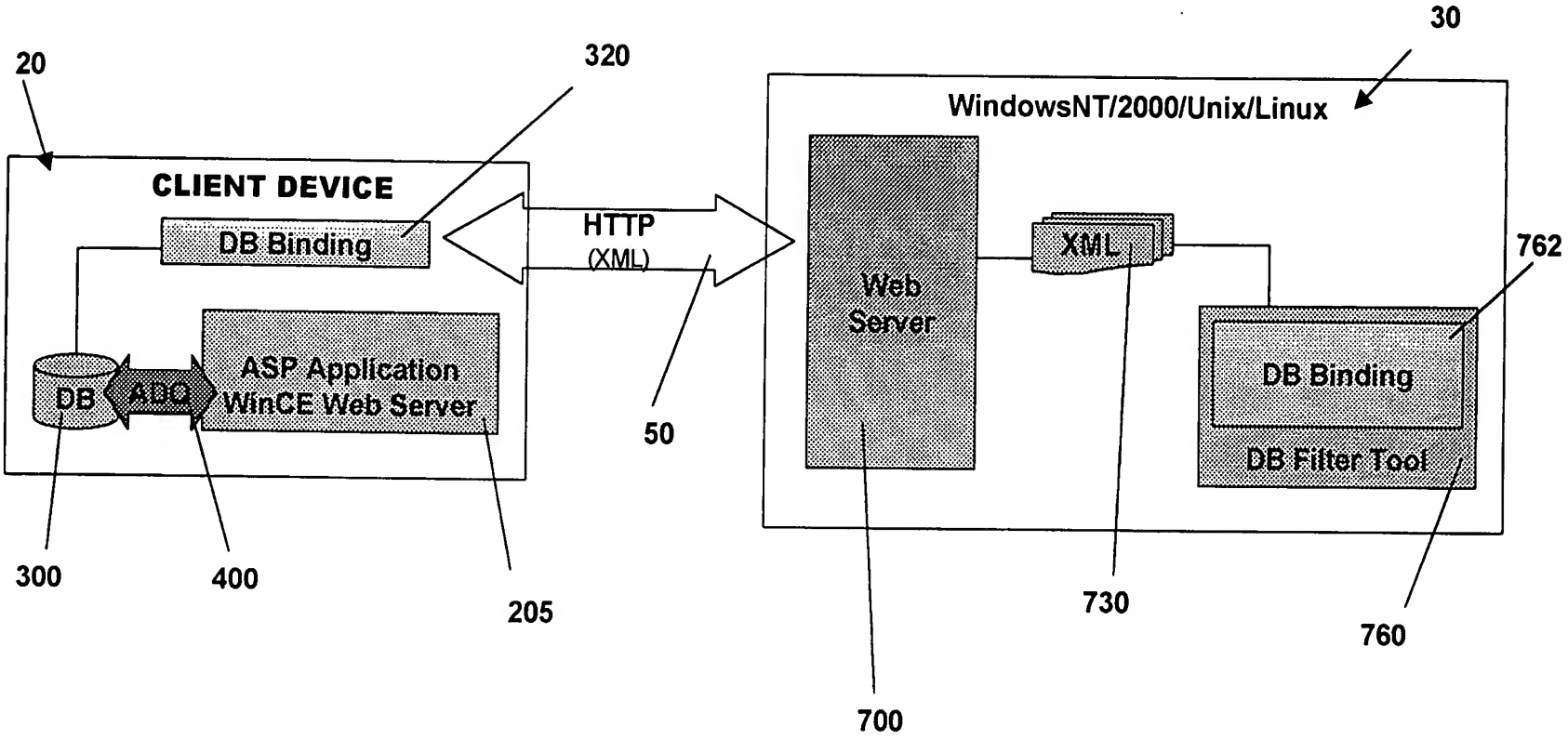

20
320
30
CLIENT DEVICE
DB Binding
HTTP
(XML)
WindowsNT/2000/Unix/Linux
Web
Server
XML
762
DB
ADO
ASP Application
WinCE Web Server
50
DB Binding
DB Filter Tool
300
400
205
730
760
700

FIGURE 9

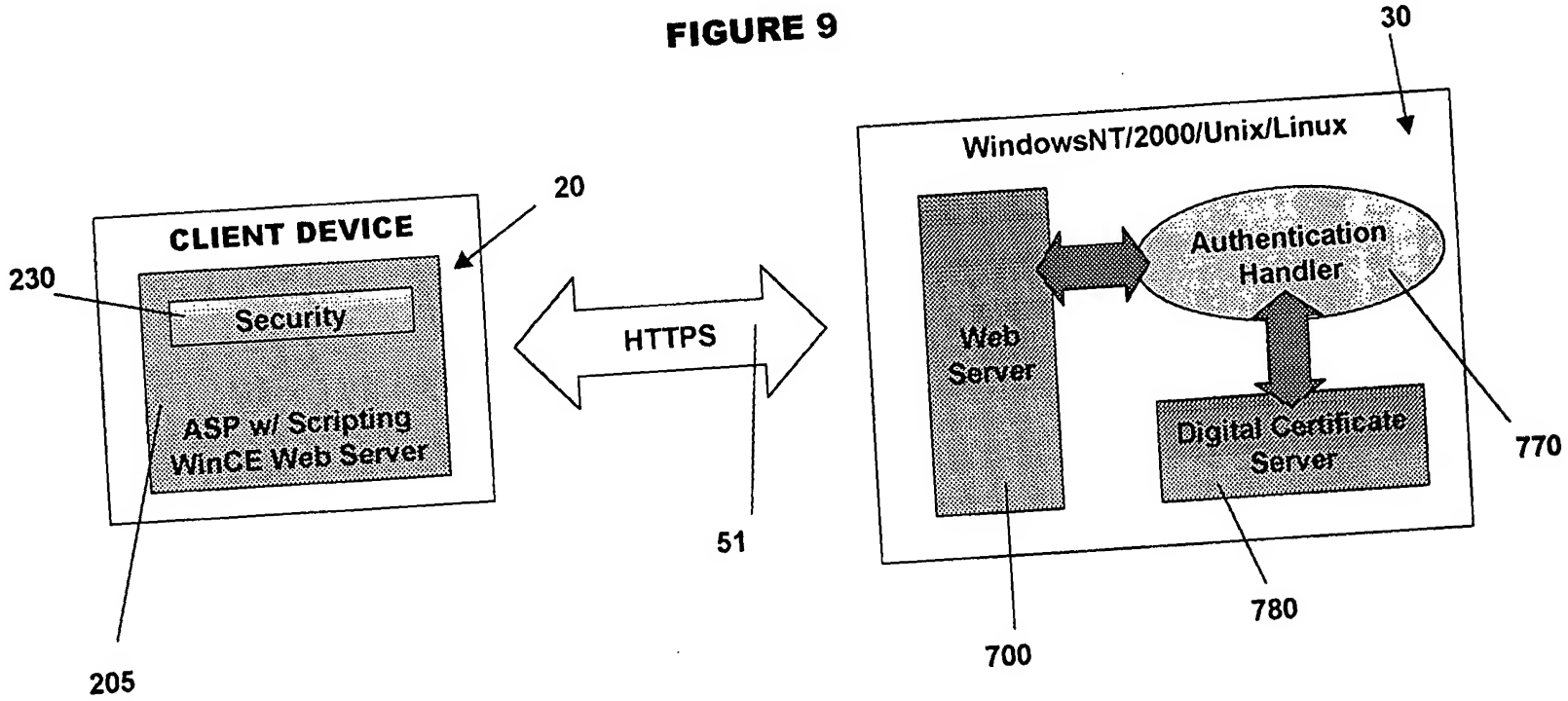

30
WindowsNT/2000/Unix/Linux
20
CLIENT DEVICE
230
Security
ASP w/ Scripting
WinCE Web Server
HTTPS
Authentication
Handler
Web
Server
Digital Certificate
Server
770
51
780
700
205

FIGURE 10

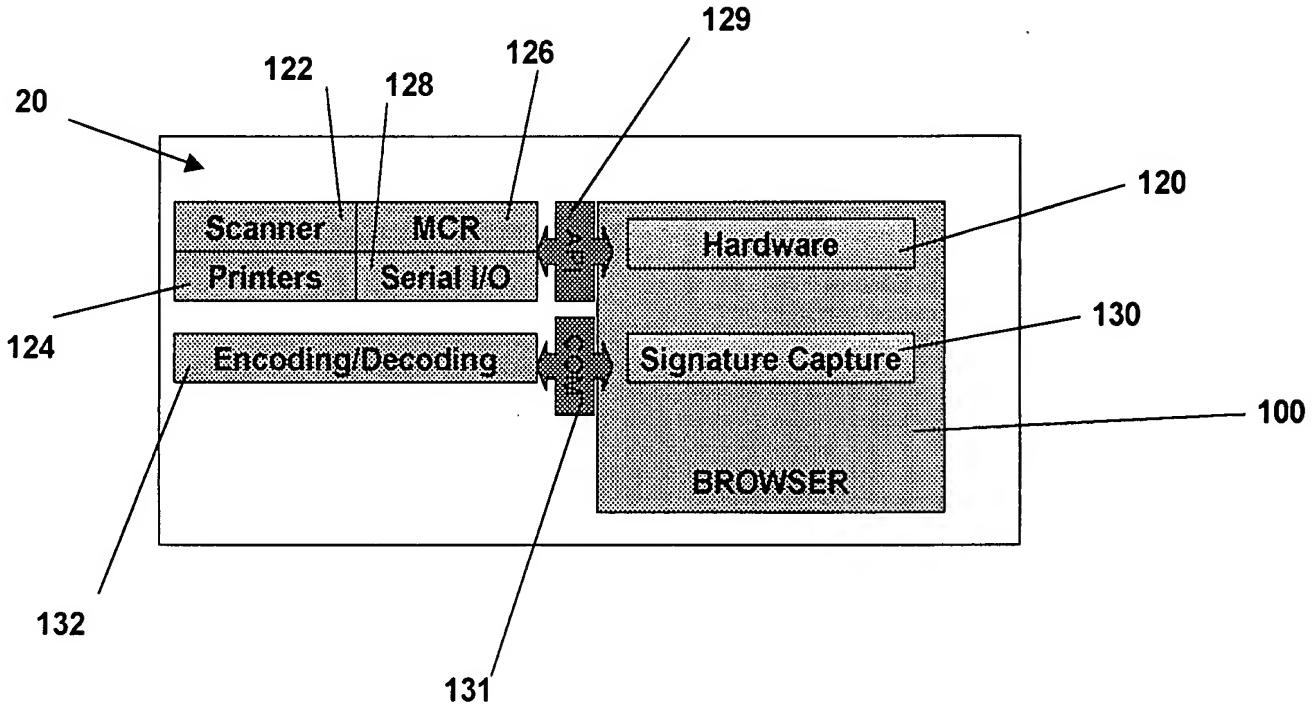

20
122
128
126
129
120
Scanner
MCR
Printers
Serial I/O
Hardware
124
130
Encoding/Decoding
Signature Capture
100
BROWSER
132
131

FIGURE 11

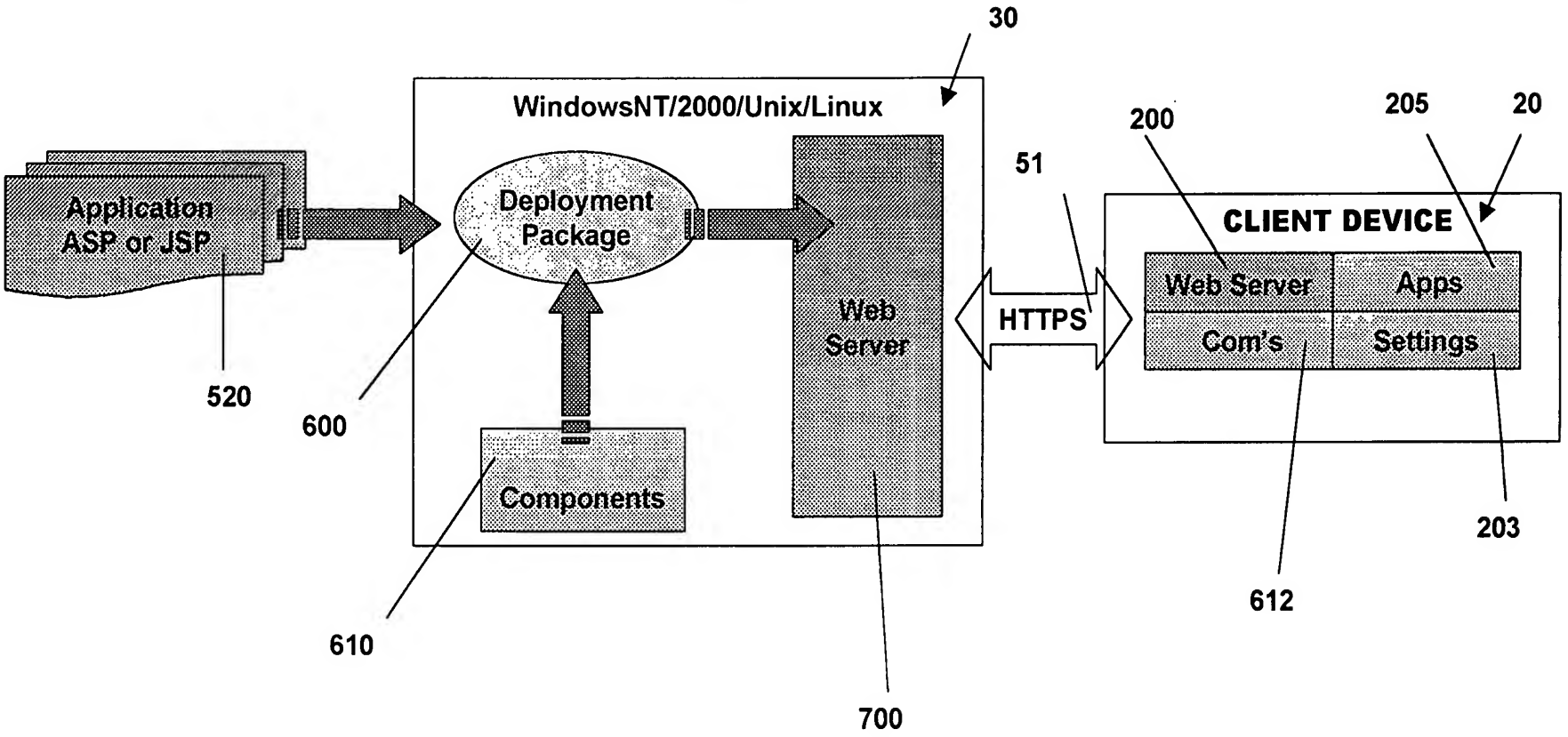

30
WindowsNT/2000/Unix/Linux
Application
ASP or JSP
Deployment
Package
Components
Web
Server
HTTPS
200
205
20
51
CLIENT DEVICE
Web Server
Apps
Com's
Settings
520
600
610
700
612
203

FIGURE 12

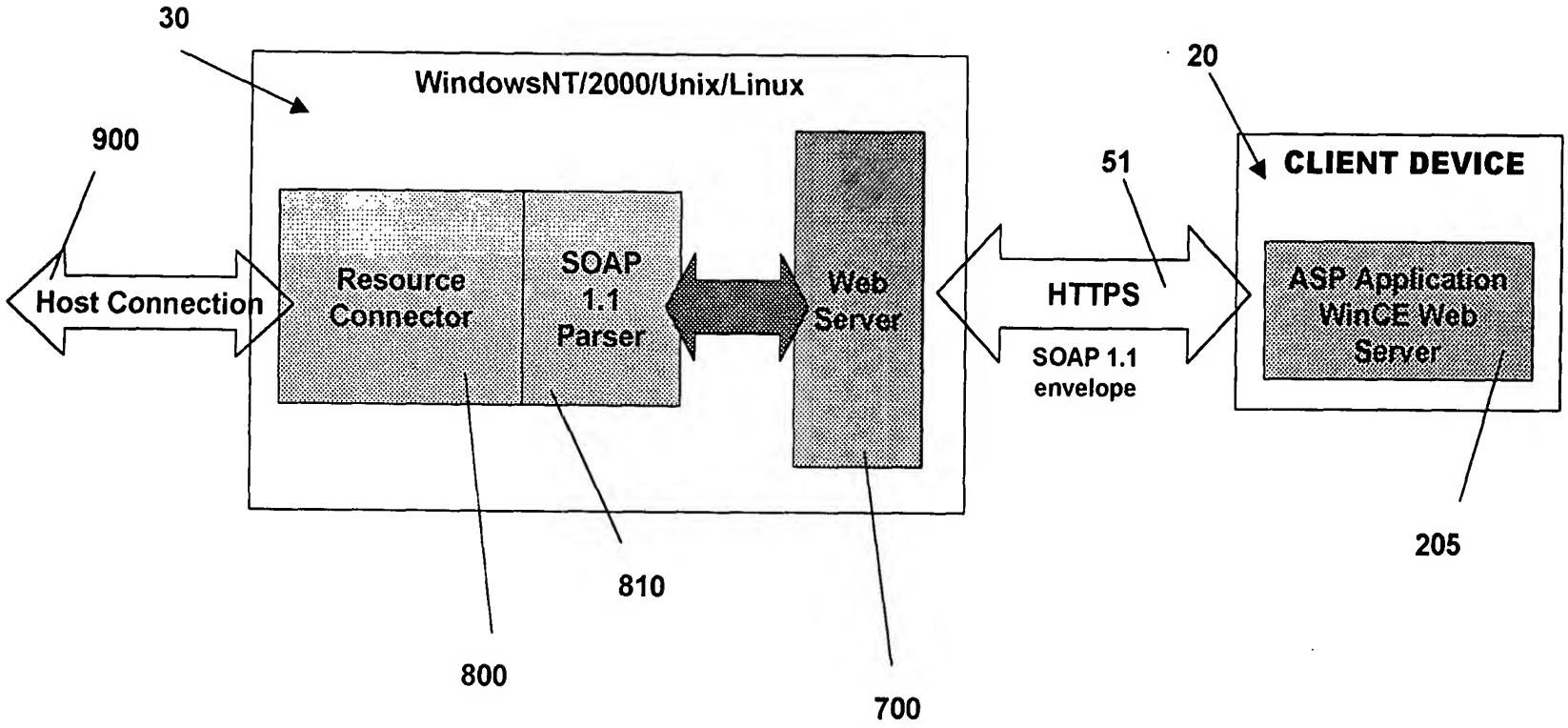
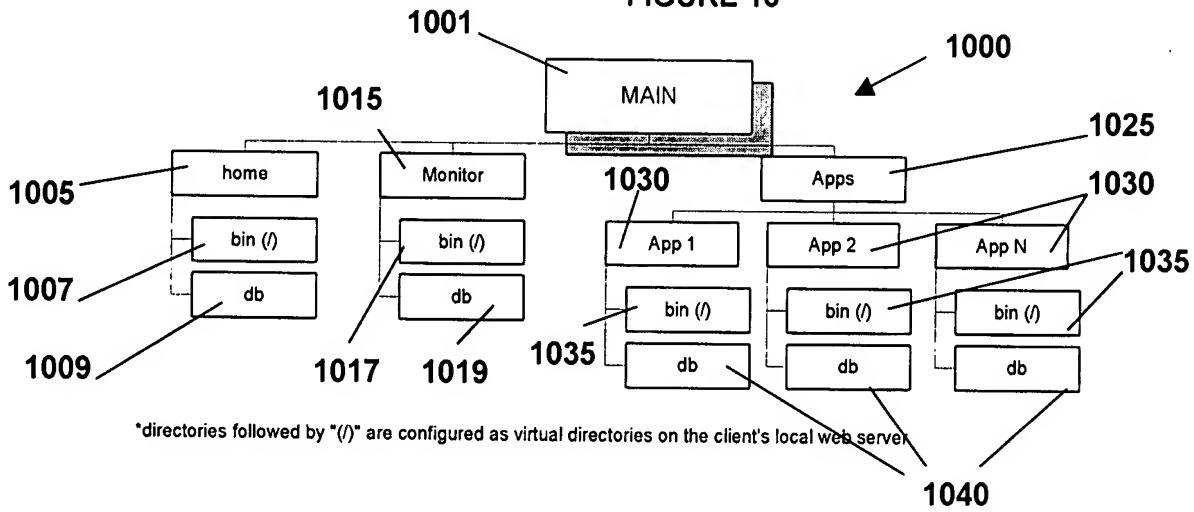

30
WindowsNT/2000/Unix/Linux
900
Host Connection
Resource Connector
SOAP 1.1 Parser
Web Server
800
810
700
51
HTTPS
SOAP 1.1 envelope
20
CLIENT DEVICE
ASP Application WinCE Web Server
205

FIGURE 13



*directories followed by "(/)" are configured as virtual directories on the client's local web server

FIGURE 14

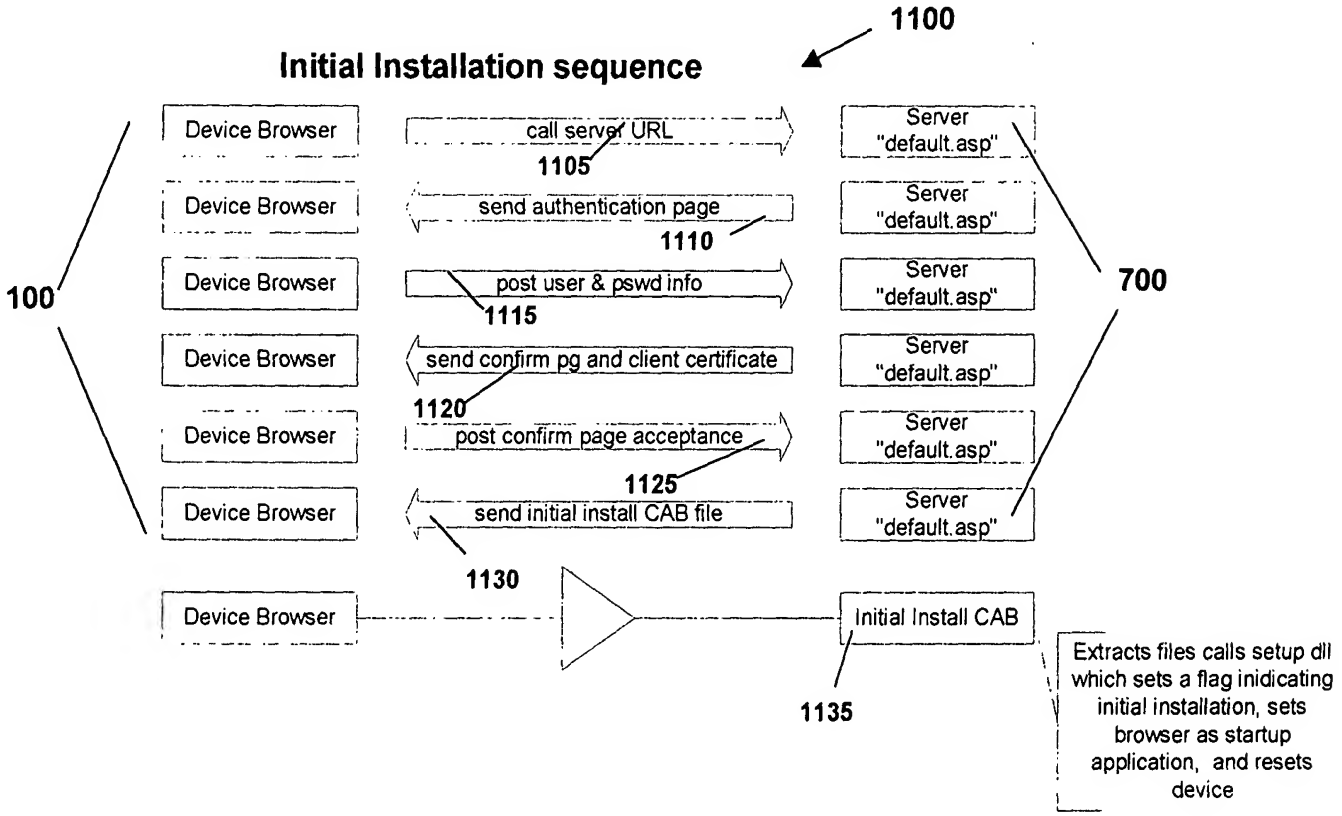

1100
Initial Installation sequence
100
Device Browser
Device Browser
Device Browser
Device Browser
Device Browser
Device Browser
call server URL
1105
send authentication page
1110
post user & pswd info
1115
send confirm pg and client certificate
1120
post confirm page acceptance
1125
send initial install CAB file
1130
Server "default.asp"
Server "default.asp"
Server "default.asp"
Server "default.asp"
Server "default.asp"
Server "default.asp"
700
Initial Install CAB
1135
Extracts files calls setup dll which sets a flag inidicating initial installation, sets browser as startup application, and resets device

FIGURE 15

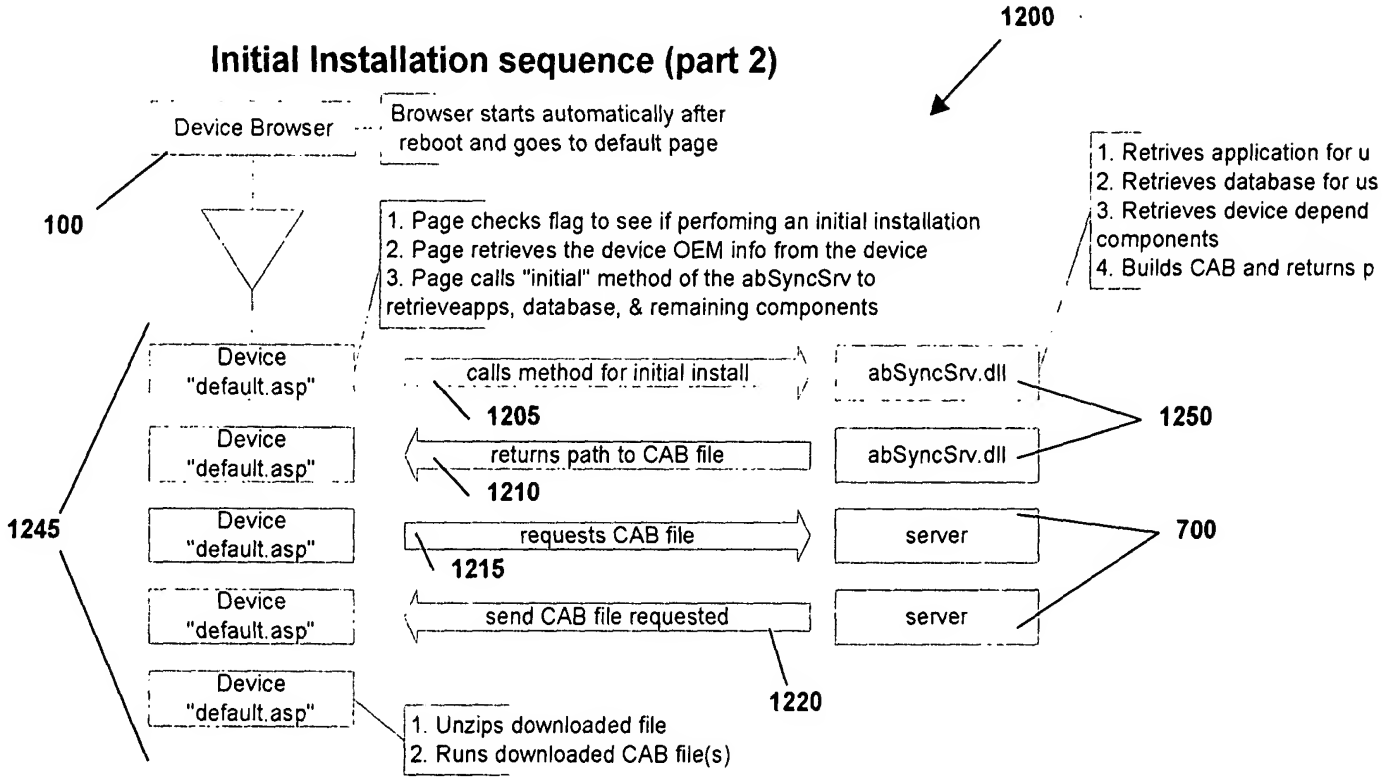

1200
Initial Installation sequence (part 2)
Device Browser
Browser starts automatically after reboot and goes to default page
100
1. Page checks flag to see if perfoming an initial installation
2. Page retrieves the device OEM info from the device
3. Page calls "initial" method of the abSyncSrv to retrieveapps, database, & remaining components
1. Retrives application for u
2. Retrieves database for us
3. Retrieves device depend components
4. Builds CAB and returns p
Device "default.asp"
calls method for initial install
abSyncSrv.dll
1205
1250
Device "default.asp"
returns path to CAB file
abSyncSrv.dll
1210
1245
Device "default.asp"
requests CAB file
server
1215
700
Device "default.asp"
send CAB file requested
server
1220
Device "default.asp"
1. Unzips downloaded file
2. Runs downloaded CAB file(s)

FIGURE 16

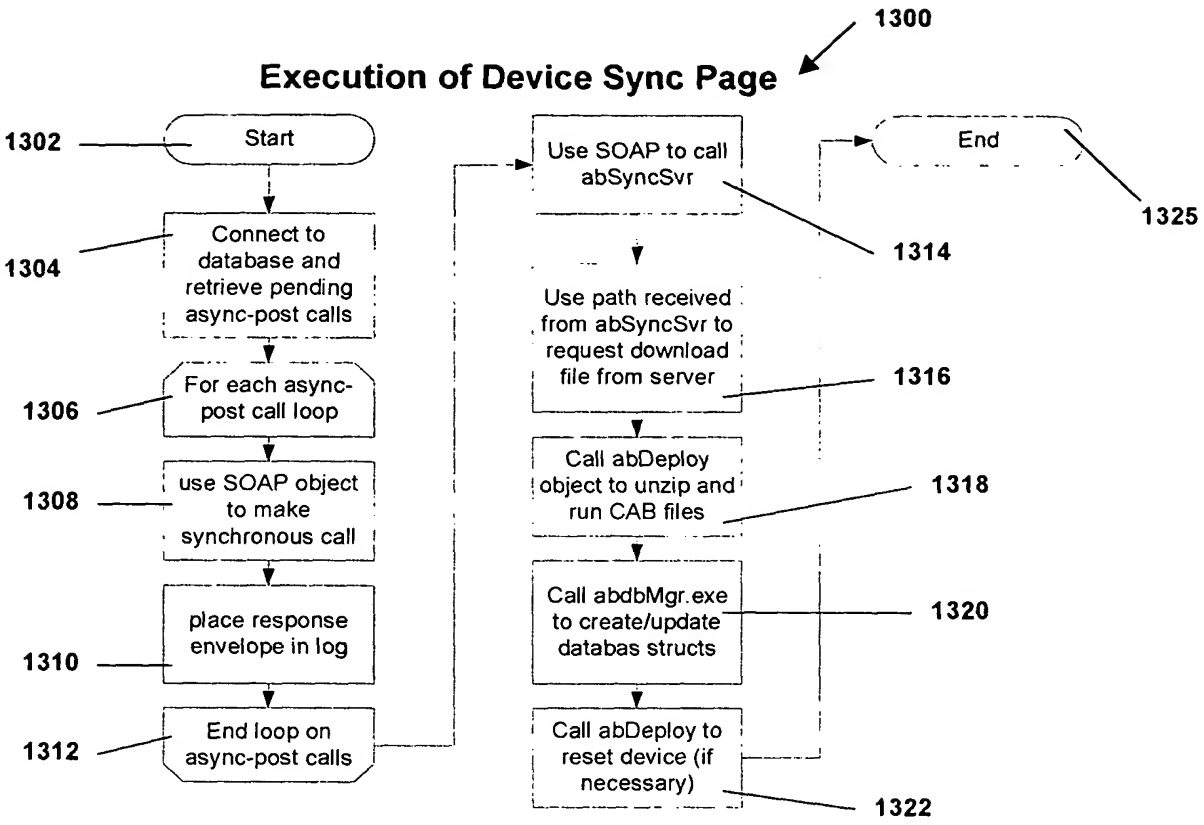

1300
Execution of Device Sync Page
1302
Start
1304
Connect to database and retrieve pending async-post calls
1306
For each async-post call loop
1308
use SOAP object to make synchronous call
1310
place response envelope in log
1312
End loop on async-post calls
Use SOAP to call abSyncSvr
1314
Use path received from abSyncSvr to request download file from server
1316
Call abDeploy object to unzip and run CAB files
1318
Call abdbMgr.exe to create/update databas structs
1320
Call abDeploy to reset device (if necessary)
1322
End
1325

FIGURE 17

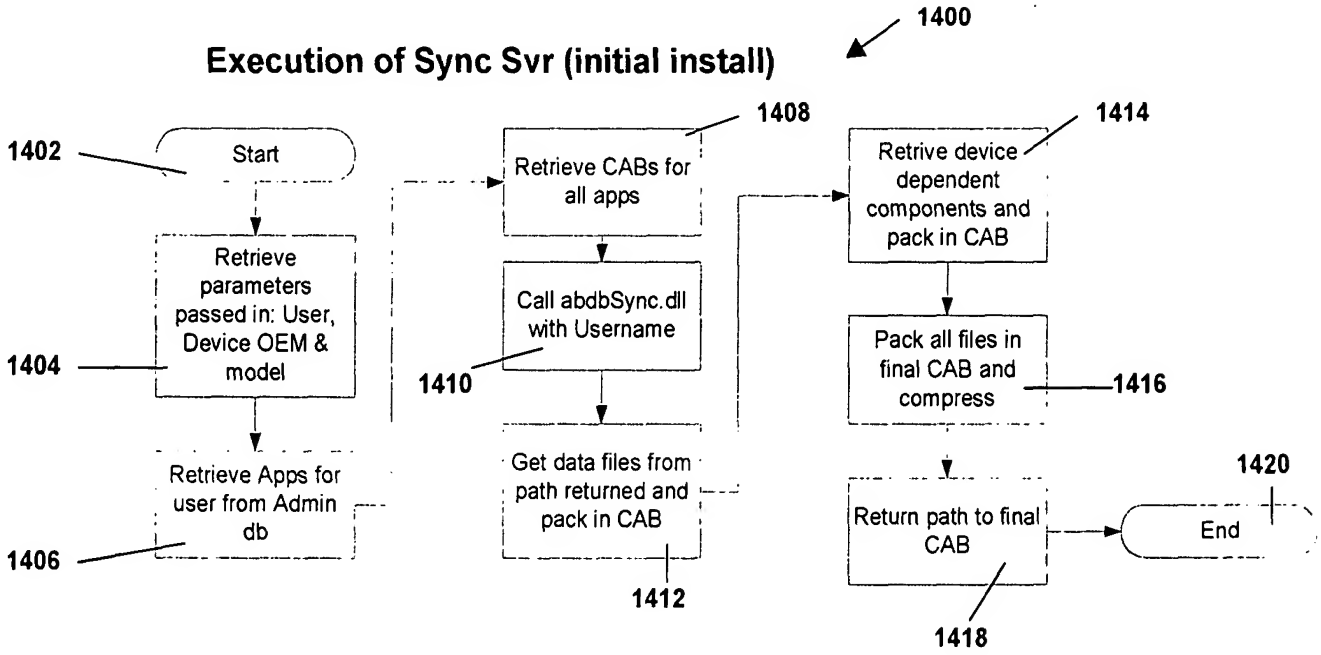

1400
Execution of Sync Svr (initial install)
1402
Start
Retrieve parameters passed in: User, Device OEM & model
1404
Retrieve Apps for user from Admin db
1406
1408
Retrieve CABs for all apps
Call abdbSync.dll with Username
1410
Get data files from path returned and pack in CAB
1412
1414
Retrive device dependent components and pack in CAB
Pack all files in final CAB and compress
1416
Return path to final CAB
1418
1420
End

FIGURE 18

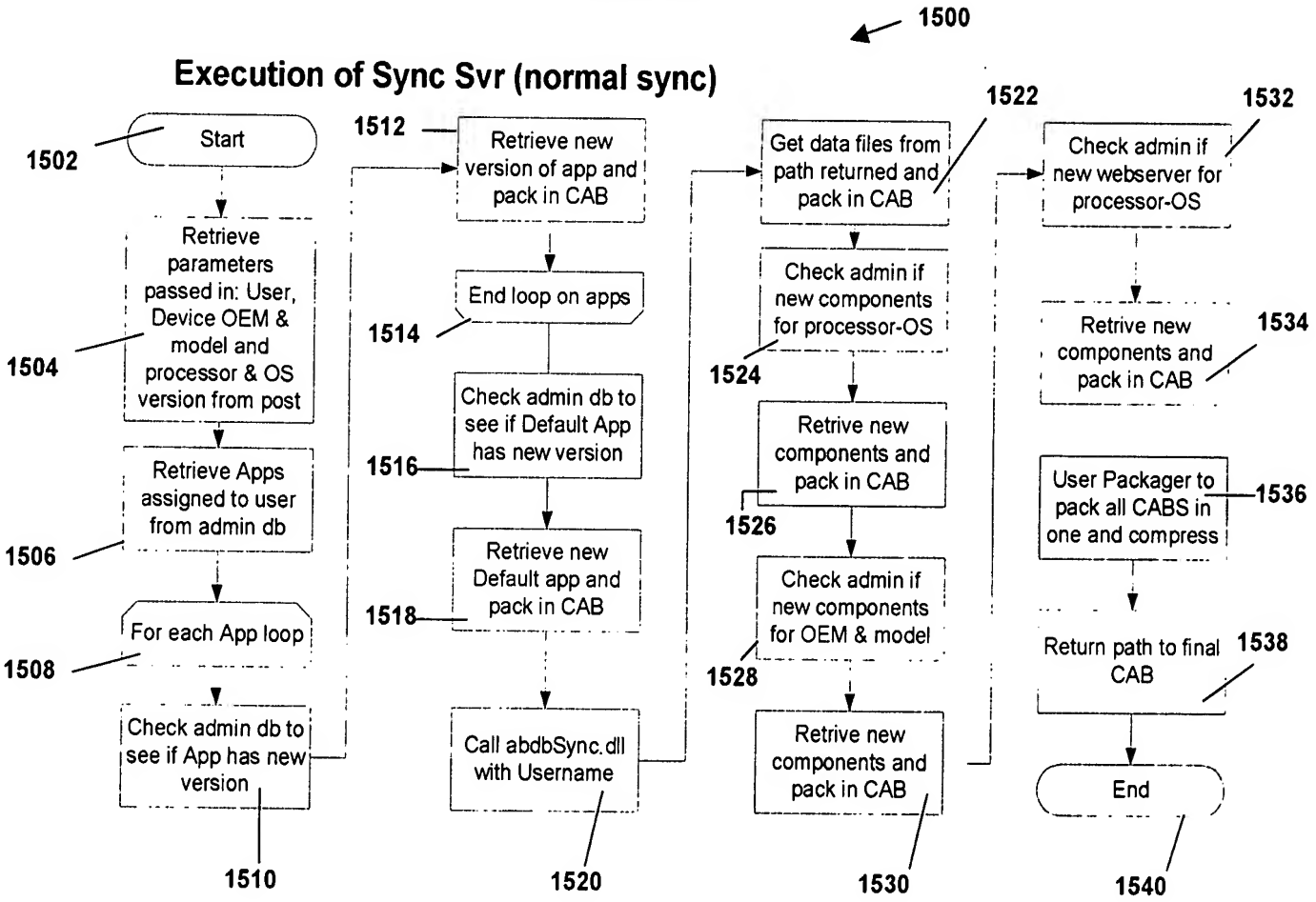

1500
Execution of Sync Svr (normal sync)
Start
1502
Retrieve parameters passed in: User, Device OEM & model and processor & OS version from post
1504
Retrieve Apps assigned to user from admin db
1506
For each App loop
1508
Check admin db to see if App has new version
1510
1512
Retrieve new version of app and pack in CAB
End loop on apps
1514
Check admin db to see if Default App has new version
1516
Retrieve new Default app and pack in CAB
1518
Call abdbSync.dll with Username
1520
1522
Get data files from path returned and pack in CAB
Check admin if new components for processor-OS
1524
Retrive new components and pack in CAB
1526
Check admin if new components for OEM & model
1528
Retrive new components and pack in CAB
1530
1532
Check admin if new webserver for processor-OS
Retrive new components and pack in CAB
1534
User Packager to pack all CABS in one and compress
1536
Return path to final CAB
1538
End
1540

FIGURE 19

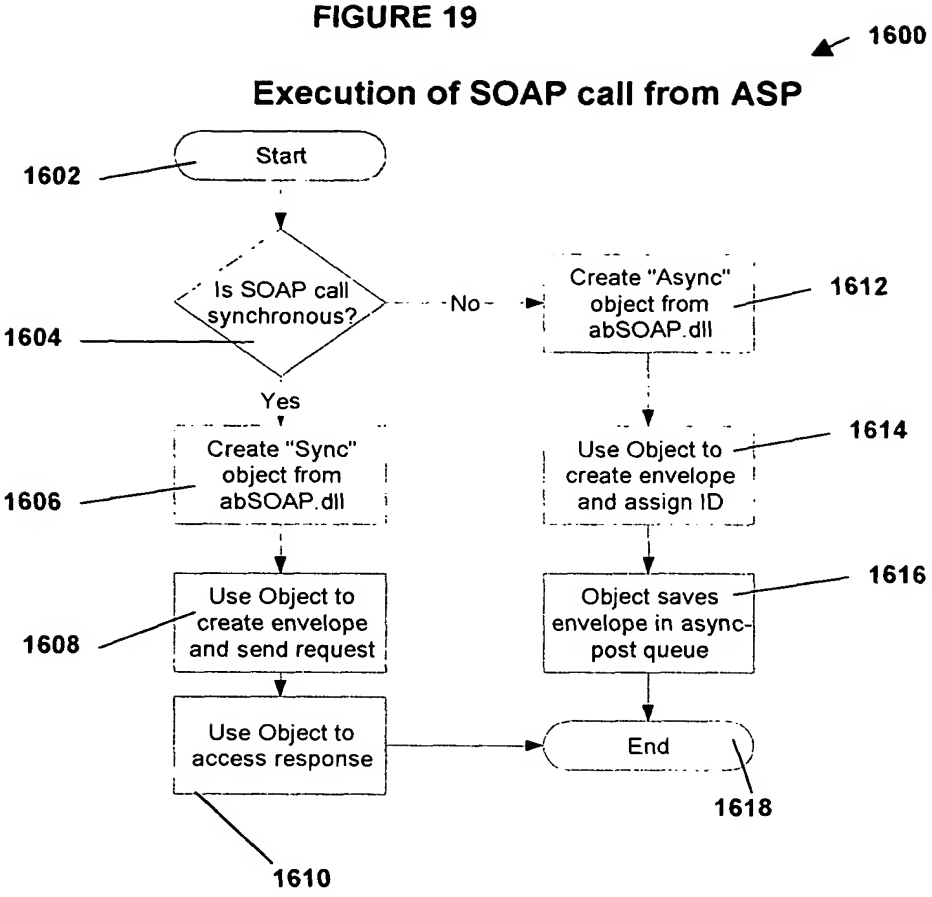
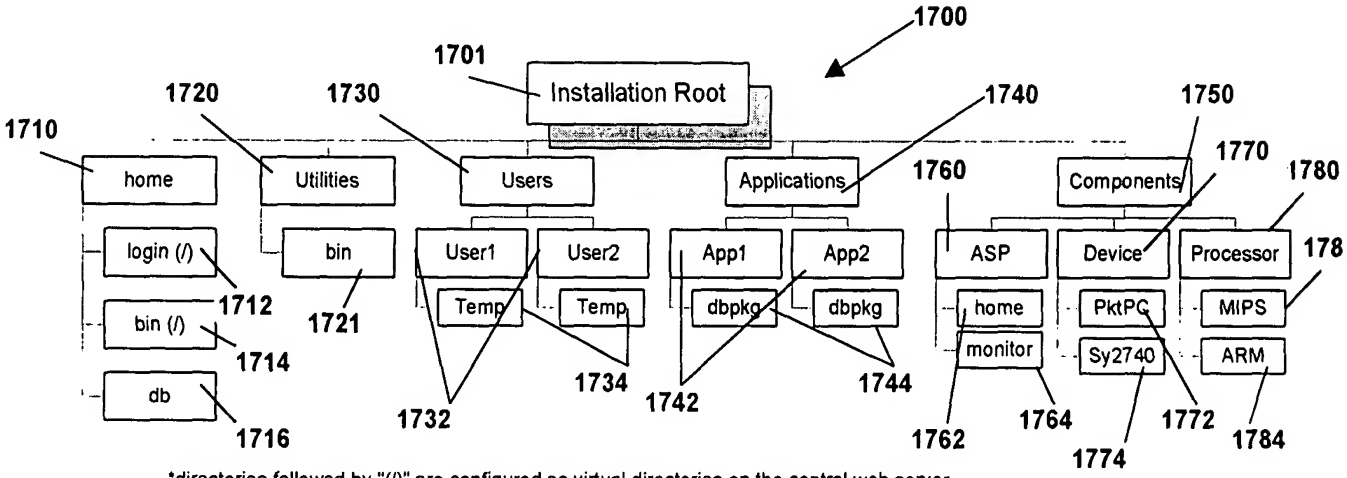
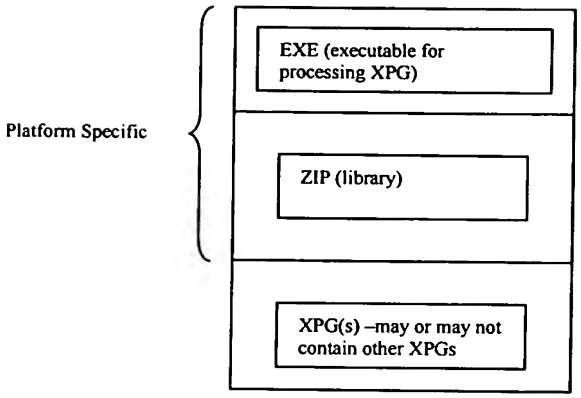

1600
Execution of SOAP call from ASP
Start
1602
Is SOAP call synchronous?
1604
No
Create "Async" object from abSOAP.dll
1612
Yes
Create "Sync" object from abSOAP.dll
1606
Use Object to create envelope and assign ID
1614
Use Object to create envelope and send request
1608
Object saves envelope in async-post queue
1616
Use Object to access response
End
1618
1610

FIGURE 20



*directories followed by "(/)" are configured as virtual directories on the central web server

FIGURE 21


Platform Specific
EXE (executable for processing XPG)
ZIP (library)
XPG(s) –may or may not contain other XPGs

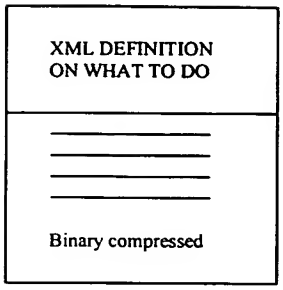



FIGURE 22